CLAIMS

What is claimed is:

1. A data scrambling method using a random data generator for a high density optical recording/reproducing apparatus using an optical disc, the data scrambling method comprising:

generating random data having a random data generation cycle based on a result by multiplying at least a size of a first data frame by a result, which is obtained by dividing a data amount of two tracks in an outermost circumference of the optical disc by a size of a second data frame.

2. The data scrambling method of claim 1, wherein the size of the first data frame is a sector, and the size of the second data frame is an error correction block.

3. The data scrambling method of claim 1, wherein the random data generation cycle is at least as great as the result obtained by multiplying at least the size of the first data frame by the result obtained by dividing the data amount of the two tracks in the outermost circumference of the optical disc by the size of the second data frame.

4. The data scrambling method of claim 1, wherein the generating of the random data comprises shift-storing, *n* bits in registers and then, generating random data, wherein a total of *n* values are used as initial values, including a first initial value, first register values, which are obtained by shifting the first initial value 7 times, a second initial value immediately after a capacity required for return of the first initial value and the first register values, and second register values, which are obtained by shifting the second initial value 7 times,

wherein the data scrambling method further comprises exclusive-ORing outputs of a predetermined number of least significant ones of the registers and input data in units of byte .

5. The data scrambling method of claim 1, wherein the generating of the random data comprises shift-storing *n* bits in registers and then, generating random data, wherein a total of n values are used as initial values, including a first initial value and register values which are supplied in each 4K times left-shifting of the first initial value;

wherein the data scrambling method further comprises exclusive-ORing outputs of a predetermined number of least significant ones of the registers and input data in units of bit.

6. The data scrambling method of claim 1, wherein the generating of the random data comprises newly setting initial values in units of an error correction block of registers generating random data in units of a sector, corresponding to 16 kinds of control values supplied in units of the error correction block,

wherein the data scrambling method further comprises exclusive-ORing outputs of a predetermined number of least significant ones of the registers and input data in units of byte.

7. The data scrambling method of claim 4, wherein:

the size of the first data frame is a sector and the size of the second data frame is an error correction block; and

the method further comprising determining the initial values by an upper 4 bits of a last byte in a 4-byte identification code which is allocated in each of a plurality of the first data frames.

8. The data scrambling method of claim 5, wherein:

the size of the first data frame is a sector and the size of the second data frame is an error correction block; and

the method further comprising determining the initial values by an upper 4 bits of a last byte in a 4-byte identification code which is allocated in each of a plurality of the first data frames.

9. The data scrambling method of claim 1, wherein the generating of the random data comprises shift-storing n bits in registers and then, generating random data, wherein a total of n values are used as initial values, including a first initial value and register values, which are supplied in each 4K left-shifting of the first initial value;

wherein the data scrambling method further comprises supplying exclusive-ORing outputs of a predetermined number of least significant ones of the registers and input data in units of byte.

10. The data scrambling method of claim 1, wherein the generating of the random data comprises:

selectively outputting n bits as valid and invalid bits in response to input m bits;

shifting and storing the n bits in serially arranged registers, to generate shifted n bits as the random data;

selecting a predetermined value or the shifted n bits for ones of the shifted n bits, to generate a selection signal; and

performing XOR operations on the ones of the shifted n bits, the ones of the shifted n bits, and an output of an adjacent more significant one of the logic circuits, and feeding back the output associated with a least significant of the ones of the shifted n bits to a least significant one of the registers.

11. The data scrambling method of claim 10, further comprising:

performing XOR operations on a plurality of least significant ones of the shifted n bits and corresponding input data bits after 8-bit left-shifting the n bits in the n registers.

12. The data scrambling method of claim 10, further comprising:

performing XOR operations on a plurality of least significant ones of the shifted n bits and corresponding input data bits after one-bit left shifting of the n bits in the n registers; and

repeating the performing of the XOR operations on the plurality of least significant ones of the shifted n bits and the corresponding input data bits after each of repeated one-bit left shifting of the n bits in the n registers.

13. The data scrambling method of claim 10, further comprising:

performing XOR operations on a plurality of least significant ones of the shifted n bits and corresponding input data bits after left shifting 4K times the n bits in the n registers.

14. A data scrambling method comprising:

scrambling data having structure of 2 KB for a sector or á data frame and 64 KB for an ECC block based on random data in a cycle of 32 KB.

15. The method of claim 14, wherein the scrambling comprises:

shifting left a 15-bit serial register r_0 through r_{14} for generating random data synchronously with a clock input for scrambling;

outputting an exclusive OR value exclusive-ORing output from the higher-most register r_{14} and output from the lower register r_{10} to the lower-most register r_0,

outputting the result of exclusive-ORing 1-byte input data D_0 through D_7 and each of the 8 outputs of the lower registers r_0 through r_7 after left-shifting the 15 bit register r_0 through r_{14} 8 times.